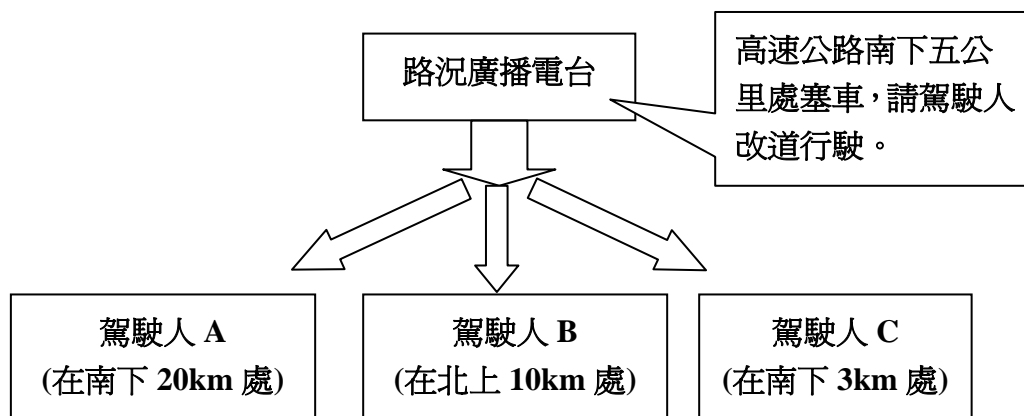


第四章 以 IPv6 為基礎之嵌入式應用架構發展實作

本章就目前車機交通路況的問題當範例，作為前面提到的 IPv6-Based Middleware 的實例，實作一個 IPv4/IPv6 車機路況即時通報系統。這章主要探討幾個部份：問題描述、系統架構、系統建置。

4.1 問題描述

許多駕駛人總是很討厭遇到塞車的情況，因此開車時會收聽路況廣播，如得知前方有塞車的情況，駕駛人可以先行改道。但如下【圖二十三】所示，路況的資訊同一份給所有的駕駛人，路況的廣播很多都不是駕駛人需要的訊息，也許我們目前的人在北部的的位置，但廣播可能報了很多南部某些地方塞車，並不符合駕駛人的需求。從【圖二十三】來看，真正需要該廣播路況資訊的人只有駕駛人 C，但駕駛人 A 與駕駛人 B 同時也都從廣播中得到該路況。



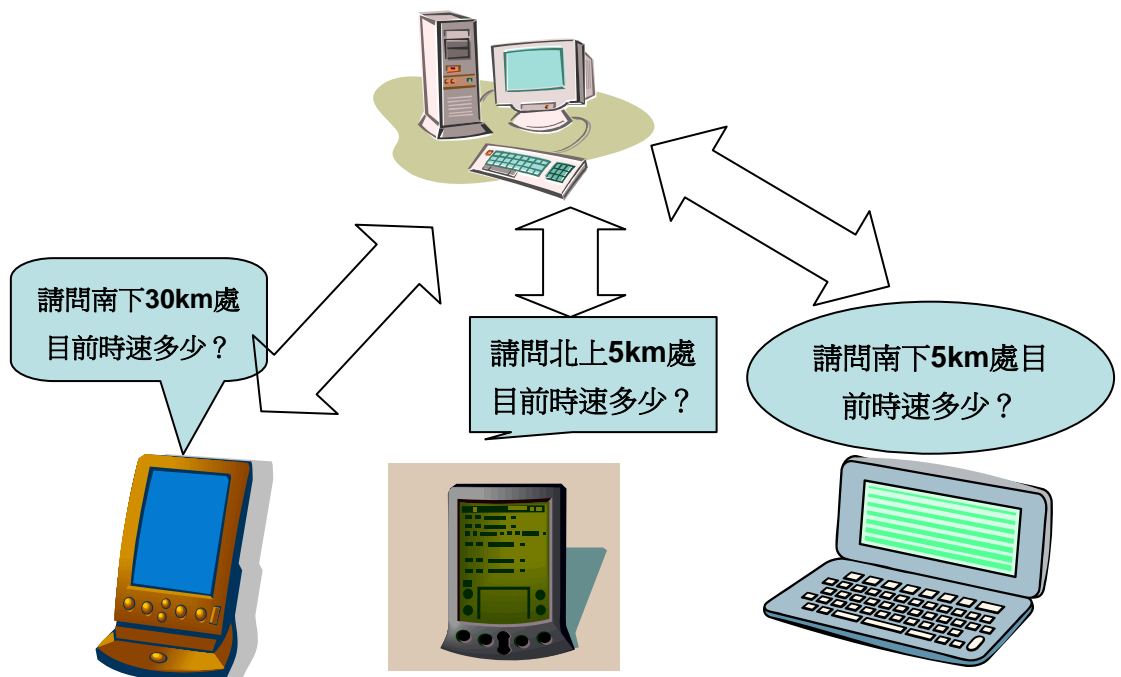
【圖二十三】 傳統路況資訊得知方式

為了避免每次得到的資訊不是自己需要的，可以把原本廣播電台傳遞資訊給我們的模式，改為如下【圖二十四】的方式。每個道路使用者，撥電話到 Call in 中心詢問自己需要的路況資訊，如此每個駕駛人可以得到自己需要的路況資訊。不過道路使用的人數眾多，因此 Call in 中心的人員勢必不堪負荷電話量，因此這個方法可行性不高。



【圖二十四】 傳統個人化路況資訊查詢方式

這個問題，若我們能把兩邊改為電腦自動處理，即可成為如下【圖二十五】的模式，提供每個使用者個人化的道路資訊，並且不會因為電話眾多無法一一接聽。



【圖二十五】 理想的路況資訊查詢方式

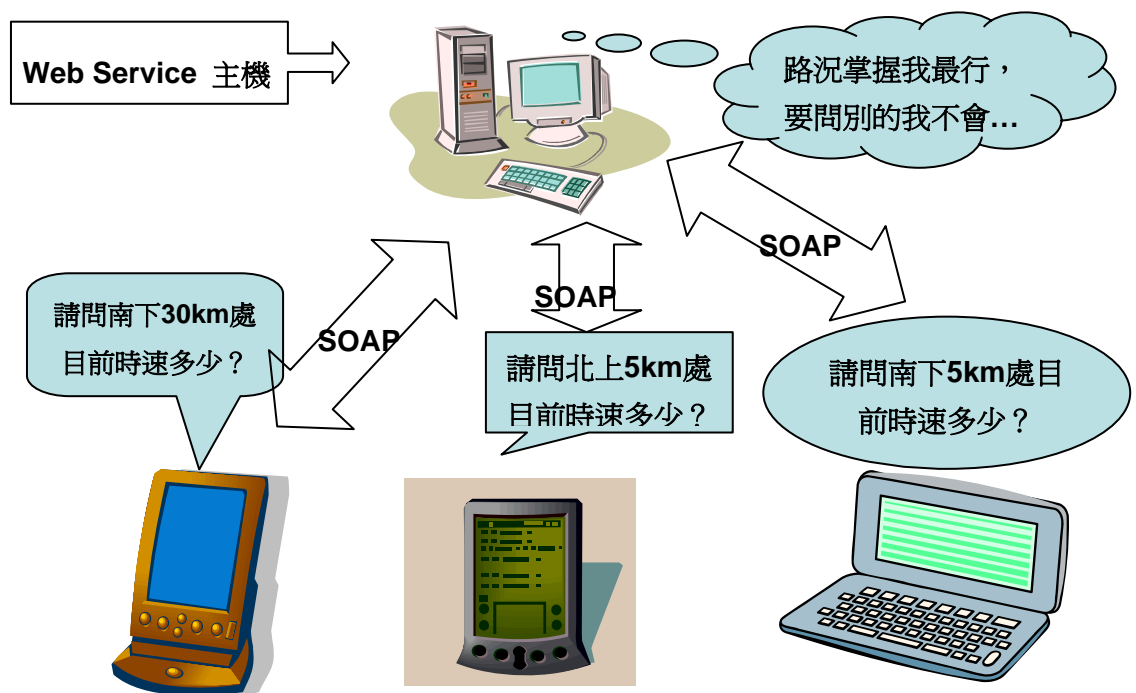
目前電腦處理速度加快，因此很多企業使用 E 化來降低企業成本，同樣的，我們也可以將原本人為接聽的方式改為電腦來處理。若使用者端能有一台裝置來做代理，跟電腦詢問個人的即時路況，不失為一個好的方法。

由第二章文獻探討可以了解，未來 IPv6 很有可能成為下一代的網路協定標準，但目前來看，IPv4 網路興盛，要從 IPv4 轉換到 IPv6 需要一段緩衝的時間，因此 IPv4 與 IPv6 有很大的機率會共存一段時間，替換成 IPv6 的時機不容易估計，或許某個應用突顯了 IPv6 與 IPv4 的不同點，就會加快更換的腳步。如果大家覺得 IPv4 與 IPv6 可以做到的應用差不多，或許就會減緩網路改為 IPv6 的時程。在目前的網路環境下情況為 IPv6 不容忽視，但 IPv4 也不能放棄的時代，因此許多目前的網際網路應用，大多希望可以做到 Dual Stack 的方式，可以同時提供 IPv4 與 IPv6 的網路服務，提高系統的可用性。

由第二章文獻探討可以發現，既有的系統因為各的格式，無法統一有個開放的溝通標準，系統之間要做整合，需要耗費很大的成本，因此多數的企業漸漸導入 Web Service 的 SOA 架構來整合企業既有的系統。用這些統一的格式來整合系統，不但降低企業成本，也提高了系統的可用性，因此遵循 SOA 系統架構應該是個趨勢。

4.2 系統架構

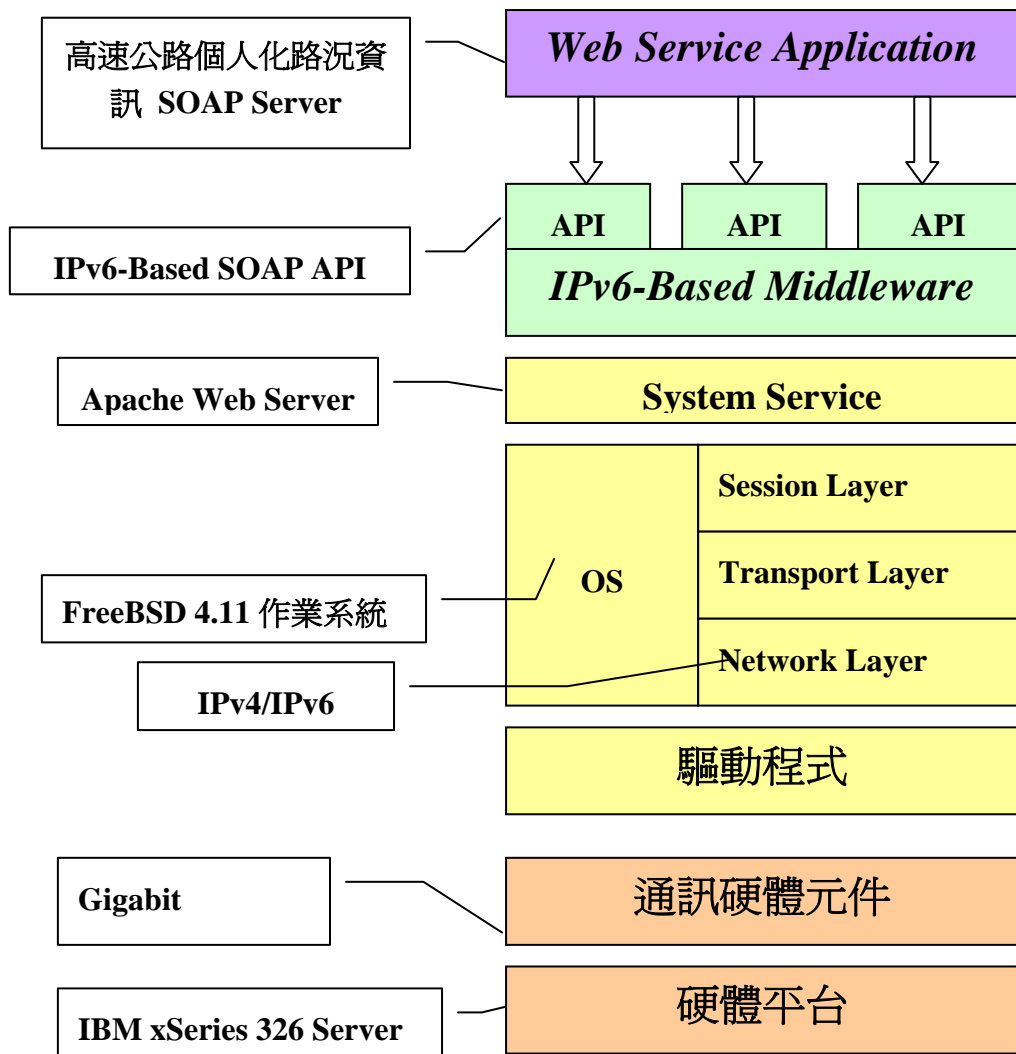
由以上需求發現，不但需要電腦處理使用者需求，使用者端最好也可以使用代理程式來做處理。網路的協定需要符合網路趨勢，因此要同時支援 IPv4 與 IPv6 網路環境。系統需使用 SOA 服務導向的架構，可以方便整合與日後增加其他服務。傳輸的內容必須加密，以提高使用的安全性。以目前電腦處理速度，以網頁伺服器來說，一台電腦可以同時處理多個需求，而不像客服人員接聽電話，每個人只能回答一個使用者的問題。本研究期待使用與網頁伺服器一樣的概念，使用 Web Service 的 SOA 架構，來製作路況資訊查詢系統，底層的通訊協定使用 IPv4 與 IPv6 共存的 Dual Stack 環境，提供查詢者個人化的交通資訊，並加上 SSL 的功能。系統的示意圖如下【圖二十六】所示。



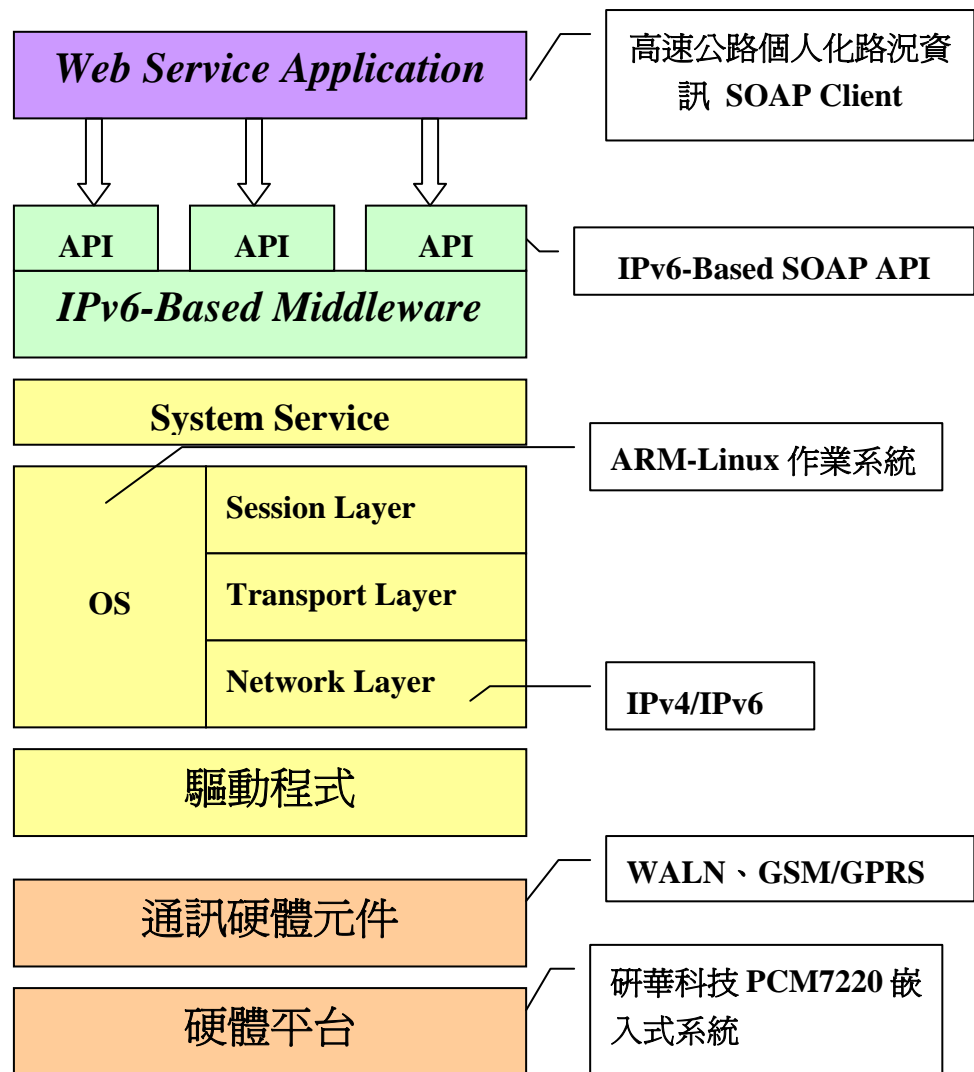
【圖二十六】 本研究設計的服務示意圖

本研究規劃開發一個 IPv6 之中介軟體使用實例，系統的伺服器端規劃如下

【圖二十七】，實作於 x86 硬體系統上。使用者端則考慮到移動性與成本低的需求，將使用範圍延伸到嵌入式系統中，該平台必須同時提供 IPv6 車機相關應用服務的執行能力。為了加速硬體實驗平台的建置時程，本研究將使用模組整合的方式進行硬體實驗平台的建置，首先選擇具有擴充能力之基礎硬體平台。本研究以研華科技 PCM-7220 為基礎硬體平台，而本研究規劃的 IPv6 中介軟體與此基礎硬體平台整合，形成 IPv6 於車機系統上的應用模式，達到駕駛人查詢即時個人化道路資訊的應用。如下【圖二十八】 系統所示，本研究現階段所規劃之車機平台將以微處理器為核心，使用 ARM-Linux 整合多項通訊模組，透過本研究之 IPv6-Based Middleware，上層的 Web Service Application 可以直接呼叫 Web Service API，不必考慮底層的網路通訊協定與硬體裝置。



【圖二十七】 系統伺服器端架構圖



【圖二十八】 系統使用者端架構圖

由以上系統架構可發現，本研究從底層硬體平台往上垂直整合到 Web Service 的功能，主要提供 IPv6-Based Middleware 與 IPv6-Based Application 的應用，以提供高速公路的即時時速服務為例子來實作。

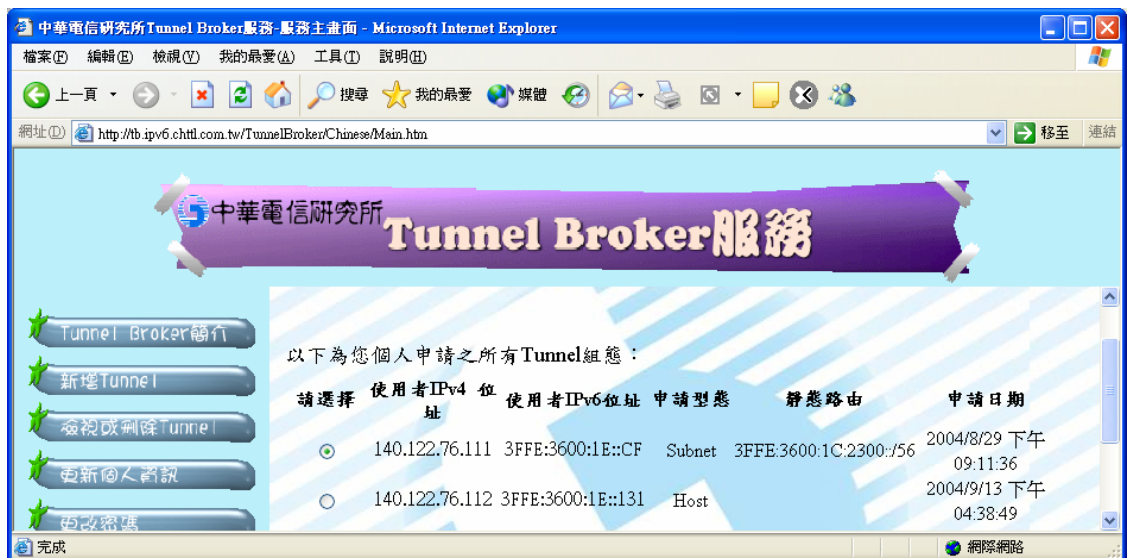
4.3 系統建置

本研究系統建置主要分為兩大階段，第一階段為從硬體建置到 IPv6 網路環境的設立，並將環境啟用為 Dual Stack，可以同時支援 IPv4 與 IPv6，第二階段為基

於 IPv6 網路環境開發出 IPv6 中介軟體，讓 Web Service 可以在 IPv6 中正常運行，並建置一個高速公路路況查詢的即時系統，讓使用者端可以透過 Web Service 的方式得知高速公路中某個位置的即時時速，查詢我們的系統得知前方道路的車速。此 IPv6 的 Web Service 平台未來仍可導入其他資訊查詢的服務，只需將資料轉為 XML 格式，透過這個平台即可應用，以下分別將這兩個階段開發的方法一一作說明。

4.3.1 第一階段開發方法

第一階段需將硬體、作業系統安裝完成，並將 IPv6 功能設定好。系統的部份，Server 可以使用 FreeBSD，安裝作業系統完成後，需將周邊的硬體驅動完成，接著啟動 FreeBSD 的 IPv6 功能，在/etc/rc.conf 加上 ipv6_enable="YES"。接著設定 IPv4 與 IPv6 Tunnel，本研究使用中華電信研究所的 Tunnel Broker，申請完成的畫面如下【圖二十九】所示：



【圖二十九】 中華電信研究所 Tunnel Broker 申請畫面

中華電信研究所與這邊設定 IPv6 Tunnel 的主機環境如下【表七】：

【表七】 IPv6 Tunnel 主機環境

| | 中華電信研究所 | 本台 FreeBSD IPv6 Router |
|--------------|------------------|------------------------|
| IPv4 位址 | 210.242.96.193 | 140.122.76.111 |
| IPv6 位址 | 3ffe:3600:1e::ce | 3FFE:3600:1E::CF |
| 分配到的 IPv6 位址 | | 3FFE:3600:1C:2300::/56 |

中華電信研究所會分配一段 IPv6 位址給我們，中華電信研究所會設定好 3FFE:3600:1C:2300::/56 這段位址的 Route 指到我們這邊，接著設定我們這端的 Tunnel，設定以上之參數，將 IPv6 封包設定 Tunnel 到中華電信研究所，設定方式在/etc/rc.conf 加上以下幾行：

```
ipv6_ifconfig_bge0="3FFE:3600:1C:2300::1 prefixlen 64"
```

```
gif_interfaces="gif0"
```

```
gifconfig_gif0="140.122.76.111 210.242.96.193"
```

```
ifconfig_gif0="inet6 3FFE:3600:1E::CF prefixlen 127"
```

```
ipv6_network_interfaces="auto"
```

```
ipv6_gateway_enable="YES"
```

```
ipv6_static_routes="default"
```

```
ipv6_route_default="default -interface gif0"
```

如果要讓整個子網路可以連上 IPv6 網路的話，可以使用 radvd(Router Advertisement Daemon)發送 router advertisement，設定 radvd 的方式在/etc/rc.conf 加上以下兩行：

```
rtadvd_enable="YES"
```

```
rtadvd_interfaces="bge0"
```

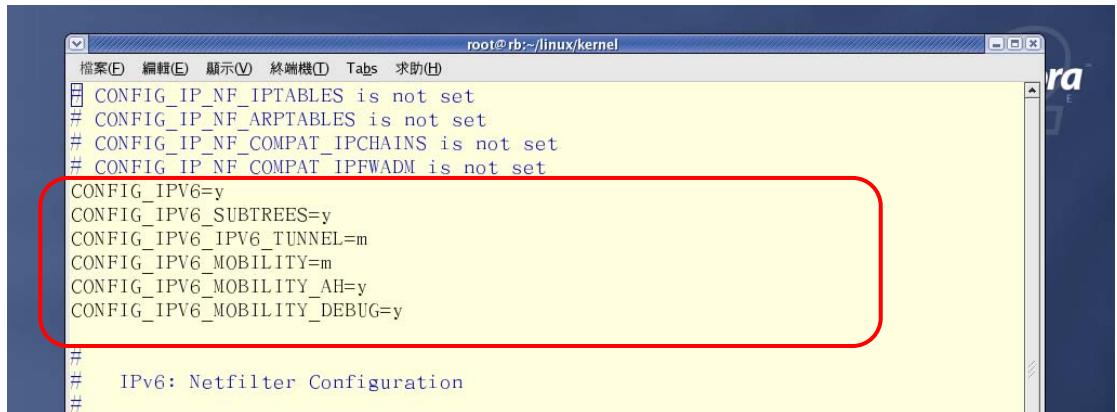
接著如果要讓其他電腦可以使用 6to4 透過這台 FreeBSD 主機連上 IPv6 網路的話，需要把 6to4 的功能打開，須在/etc/rc.conf 加入以下兩行：

```
stf_interface_ipv4addr="140.122.76.111"
```

```
stf_interface_ipv4plen="16"
```

增加的 6to4 的功能後，140.122.xxx.xxx 的電腦可以設定 6to4 的方式透過這台主機連上 IPv6 網路。

嵌入式系統的部份，選擇安裝 ARM-Linux 嵌入式作業系統，將硬體驅動完成後，修改紀錄 kernel 選項的檔案.config，如下【圖三十】所示：

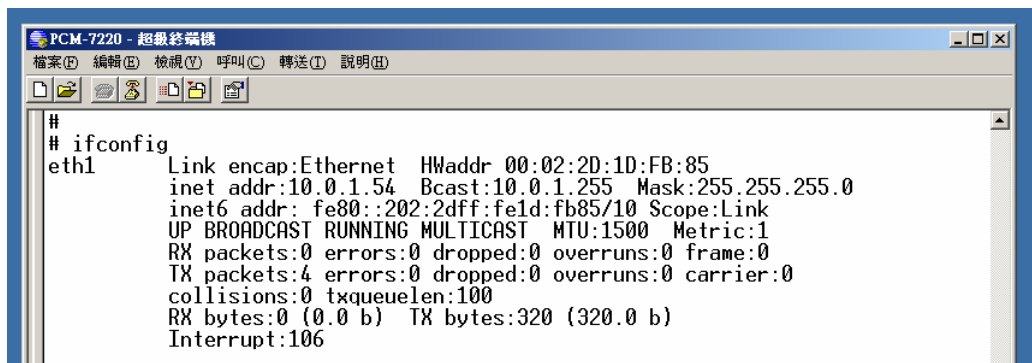


```
root@rb:~/linux/kernel
CONFIG_IP_NF_IPTABLES is not set
# CONFIG_IP_NF_ARPTABLES is not set
# CONFIG_IP_NF_COMPAT_IPCHAINS is not set
# CONFIG_IP_NF_COMPAT_IPFWADM is not set
CONFIG_IPV6=y
CONFIG_IPV6_SUBTREES=y
CONFIG_IPV6_IPV6_TUNNEL=m
CONFIG_IPV6_MOBILITY=m
CONFIG_IPV6_MOBILITY_AH=y
CONFIG_IPV6_MOBILITY_DEBUG=y
#
# IPv6: Netfilter Configuration
#
```

【圖三十】 ARM-Linux 嵌入式作業系統啟動 IPv6 功能設定

設定完成後重新編譯 kernel 並複製到嵌入式系統執行，嵌入式系統重新開機

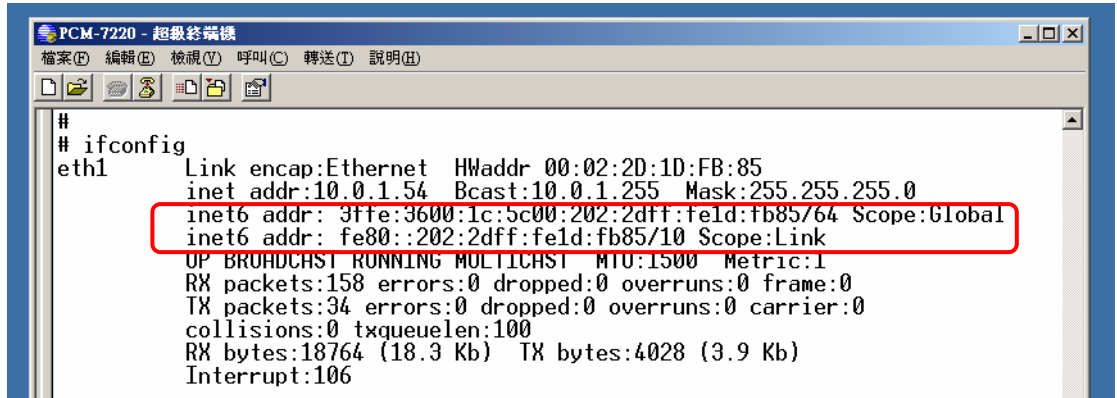
後即擁有 IPv6 的功能如下【圖三十一】所示：



```
PCM-7220 - 超級終端機
# ifconfig
eth1  Link encap:Ethernet  Hwaddr 00:02:2D:1D:FB:85
      inet addr:10.0.1.54  Bcast:10.0.1.255  Mask:255.255.255.0
      inet6 addr: fe80::202:2dff:fe1d:fb85/10  Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      RX bytes:0 (0.0 b)  TX bytes:320 (320.0 b)
      Interrupt:106
```

【圖三十一】 ARM-Linux 嵌入式作業系統啟動 IPv6 功能結果

完成之後可使用 Infrastructure 模式連上與 IPv6 Router 同一個子網路的無線 AP，或者使用 Ad hoc 模式連上 IPv6 Router 的無線網卡，收到 router advertisement 透過 IPv6 Router 連接到 IPv6 網路。若有成功可以發現嵌入式系統取得 IPv6 位址如下【圖三十二】所示：

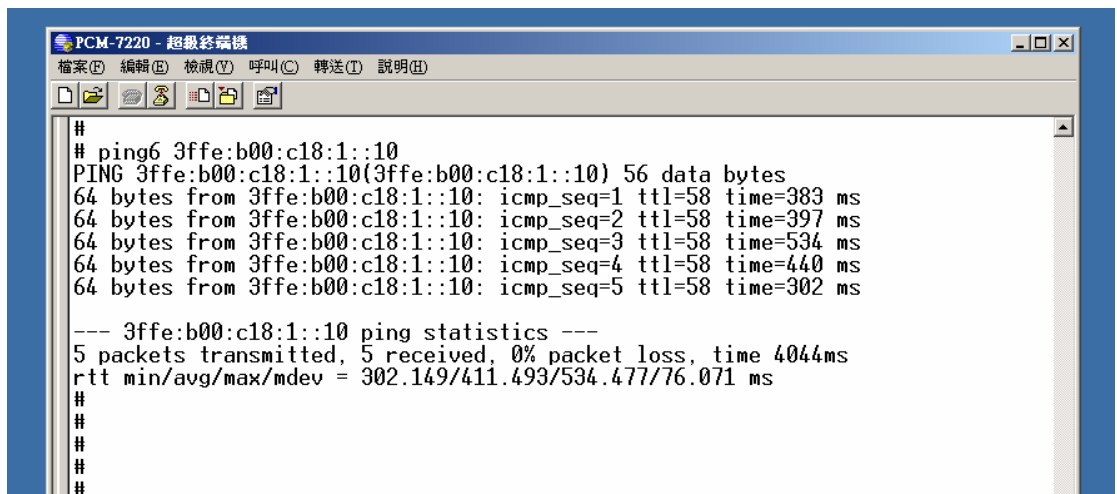


```
PCM-7220 - 超級終端機
檔案(F) 編輯(E) 檢視(V) 呼叫(C) 轉送(T) 說明(H)

#
# ifconfig
eth1      Link encap:Ethernet  HWaddr 00:02:2D:1D:FB:85
          inet addr:10.0.1.54  Bcast:10.0.1.255  Mask:255.255.255.0
          inet6 addr: 3ffe:3600:1c:5c00:202:2dff:fe1d:fb85/64 Scope:Global
          inet6 addr: fe80::202:2dff:fe1d:fb85/10 Scope:Link
          UP BRUHDCHST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:158 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:18764 (18.3 Kb)  TX bytes:4028 (3.9 Kb)
          Interrupt:106
```

【圖三十二】 ARM-Linux 嵌入式作業系統取得 IPv6 位址畫面

接著可以測試使用 ping6 指令到一個存在的 IPv6 位址，看看是否有正常回應，若可以正常回應的話會得到類似如下【圖三十三】的畫面：



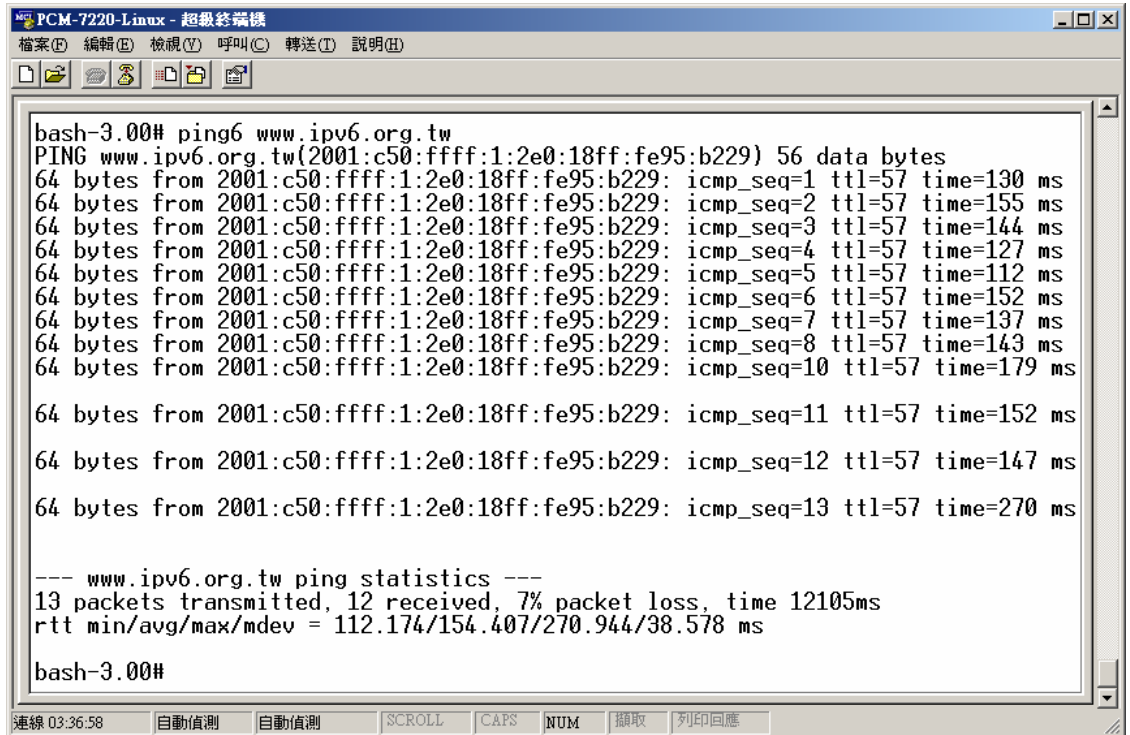
```
PCM-7220 - 超級終端機
檔案(F) 編輯(E) 檢視(V) 呼叫(C) 轉送(T) 說明(H)

#
# ping6 3ffe:b00:c18:1::10
PING 3ffe:b00:c18:1::10(3ffe:b00:c18:1::10) 56 data bytes
64 bytes from 3ffe:b00:c18:1::10: icmp_seq=1 ttl=58 time=383 ms
64 bytes from 3ffe:b00:c18:1::10: icmp_seq=2 ttl=58 time=397 ms
64 bytes from 3ffe:b00:c18:1::10: icmp_seq=3 ttl=58 time=534 ms
64 bytes from 3ffe:b00:c18:1::10: icmp_seq=4 ttl=58 time=440 ms
64 bytes from 3ffe:b00:c18:1::10: icmp_seq=5 ttl=58 time=302 ms

--- 3ffe:b00:c18:1::10 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4044ms
rtt min/avg/max/mdev = 302.149/411.493/534.477/76.071 ms
#
#
#
#
```

【圖三十三】 ARM-Linux 嵌入式作業系統連上 IPv6 畫面

接著可以在嵌入式系統的/etc 目錄加入 resolv.conf，讓嵌入式系統可以查詢 DNS 主機，找出 Domain Name 與 IP 的對應，如此可以不用記憶 IPv6 冗長的 IP 位址。設定之後可以 ping IPv6 主機名稱，連上畫面如下【圖三十四】所示：



```
bash-3.00# ping6 www.ipv6.org.tw
PING www.ipv6.org.tw(2001:c50:ffff:1:2e0:18ff:fe95:b229) 56 data bytes
64 bytes from 2001:c50:ffff:1:2e0:18ff:fe95:b229: icmp_seq=1 ttl=57 time=130 ms
64 bytes from 2001:c50:ffff:1:2e0:18ff:fe95:b229: icmp_seq=2 ttl=57 time=155 ms
64 bytes from 2001:c50:ffff:1:2e0:18ff:fe95:b229: icmp_seq=3 ttl=57 time=144 ms
64 bytes from 2001:c50:ffff:1:2e0:18ff:fe95:b229: icmp_seq=4 ttl=57 time=127 ms
64 bytes from 2001:c50:ffff:1:2e0:18ff:fe95:b229: icmp_seq=5 ttl=57 time=112 ms
64 bytes from 2001:c50:ffff:1:2e0:18ff:fe95:b229: icmp_seq=6 ttl=57 time=152 ms
64 bytes from 2001:c50:ffff:1:2e0:18ff:fe95:b229: icmp_seq=7 ttl=57 time=137 ms
64 bytes from 2001:c50:ffff:1:2e0:18ff:fe95:b229: icmp_seq=8 ttl=57 time=143 ms
64 bytes from 2001:c50:ffff:1:2e0:18ff:fe95:b229: icmp_seq=10 ttl=57 time=179 ms

64 bytes from 2001:c50:ffff:1:2e0:18ff:fe95:b229: icmp_seq=11 ttl=57 time=152 ms

64 bytes from 2001:c50:ffff:1:2e0:18ff:fe95:b229: icmp_seq=12 ttl=57 time=147 ms

64 bytes from 2001:c50:ffff:1:2e0:18ff:fe95:b229: icmp_seq=13 ttl=57 time=270 ms

--- www.ipv6.org.tw ping statistics ---
13 packets transmitted, 12 received, 7% packet loss, time 12105ms
rtt min/avg/max/mdev = 112.174/154.407/270.944/38.578 ms
bash-3.00#
```

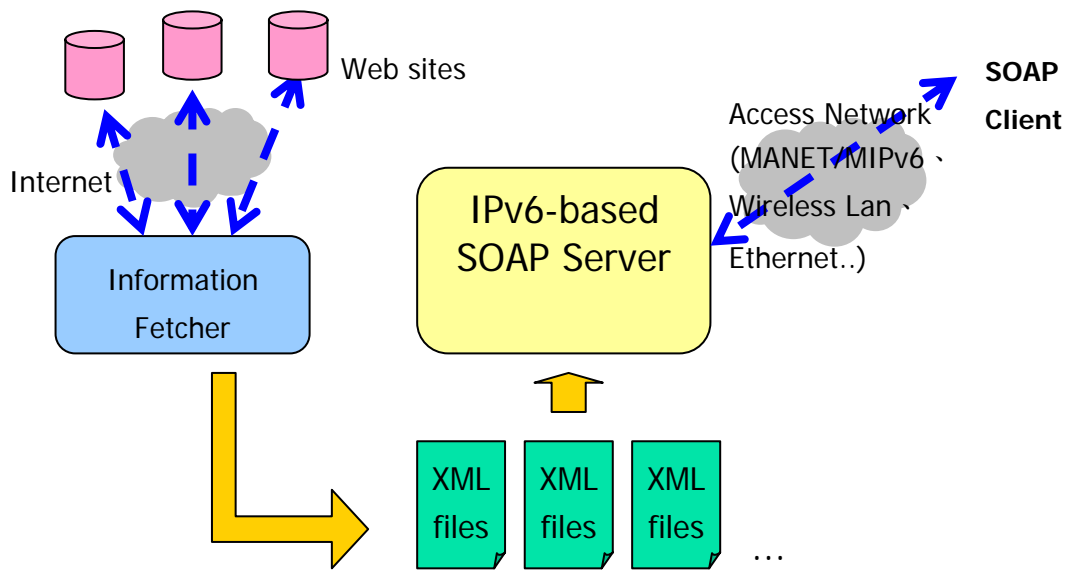
【圖三十四】 ARM-Linux 嵌入式系統用 Domain Name 連上 IPv6 畫面

如此，伺服器端與用戶端都已經可以正常連上 IPv6 網路，接著可以繼續第二階段導入 IPv6 通訊協定為基礎之 SOAP 訊息技術。

4.3.1 第二階段開發方法

第二階段將 SOAP 訊息技術導入第一階段的 IPv6 網路環境中，系統規劃模式

如下【圖三十五】：



【圖三十五】 系統規劃模式

Information Fetcher 到網站上抓取 SOAP 服務程式需要的資料轉為 XML 檔案，SOAP Server 與 SOAP Client 之間使用無線傳輸資料，SOAP Server 依據 XML 檔案的內容，回覆 Client 需要的資訊。本研究實作上依據這樣的架構，Information Fetcher 到網站抓取高路公路的即時時速轉為 XML 檔案，SOAP Client 使用嵌入式系統傳遞所要詢問時速的高速公路位置，SOAP Server 傳回該區域的即時時速。

XML 檔案使用撰寫的 Parser 程式將高速公路時速轉為 XML 檔案，XML 檔案內主要包含高速公路某一路段到某一路段的時速是多少。例如：從 9.4 公里到 10.5 公里處南下方向時速 50 公里，北上方向時速 80 公里。一個簡單有三筆資料的高速公路即時時速之 XML 檔案如下【表八】所示：

【表八】 簡單的高速公路時速資訊 XML 檔案

```

<?xml version="1.0" encoding="utf-8"?>
<freeway>
  <area no="1" from="0.0" to="1.1" title="基隆端 - 基隆交流道">
    <south>30</south>
    <north>42</north>
  </area>
  <area no="2" from="1.1" to="2.5" title="基隆交流道 - 八堵交流道">
    <south>560</south>
    <north>63</north>
  </area>
  <area no="3" from="2.5" to="6.8" title="八堵交流道 - 五堵交流道">

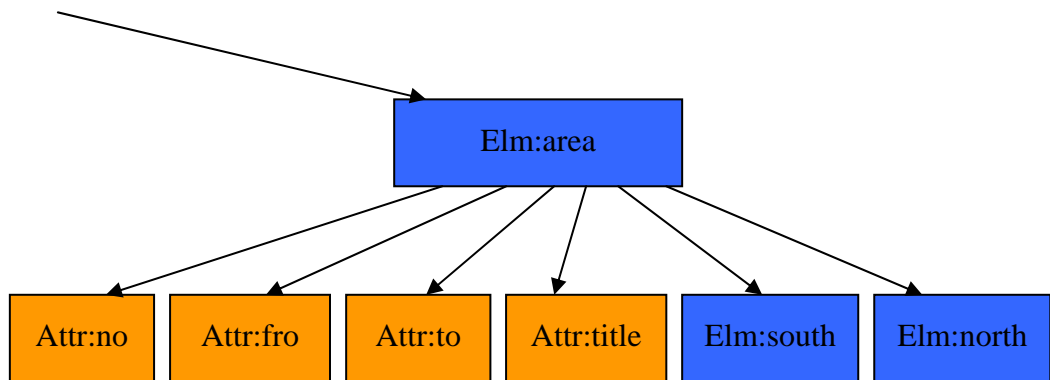
```

```

    <south>83</south>
    <north>74</north>
  </area>
</freeway>

```

上表中的 XML 檔案中”no”、”from”、”to”和”title”是”area”的屬性，”south”與”north”是”area”的子元素，這些子元素和屬性可以表示如下【圖三十六】所示：



【圖三十六】 高速公路即時時速 XML 檔案架構

若以表格的概念來看，XML 檔案也可以使用 XSL 語法將上面三筆資料的 XML 檔案轉為表格的呈現方式，【表八】的 XML 檔案用 XSL 語法轉換後結果如下【表九】所示：（XSL 檔案如附錄四）

【表九】以 XSL 用表格方式呈現簡單的高速公路時速資訊

| 編號 | 起始里程 | 結束里程 | 位置 | 南下時速 | 北上時速 |
|----|------|------|---------------|------|------|
| 1 | 0.0 | 1.1 | 基隆端 - 基隆交流道 | 30 | 42 |
| 2 | 1.1 | 2.5 | 基隆交流道 - 八堵交流道 | 56 | 63 |
| 3 | 2.5 | 6.8 | 八堵交流道 - 五堵交流道 | 83 | 74 |

本 XML 的 Schema 如下【圖三十七】所示，freeway 底下有一到無限多個 area，每個 area 裡面有該區域的南下時速(south)與北上時速(north)。



【圖三十七】 高速公路即時時速 XML 檔案 Schema 圖

接著 SOAP Server 的平台在第一階段開發後已經具備 IPv6 的通訊功能，接著將系統加入 Apache 網頁伺服器套件，讓 SOAP 可以跑在 Apache 上，再將 Libxml2 與 openssl 兩個套件安裝到系統中。FreeBSD 系統可以使用 ports 套件管理來安裝，Linux 主機可以使用 rpm 指令安裝。完成後將 OpenSOAP 安裝於系統中，預設的安裝目錄為 /usr/local/opensoap，由於 Server 端為一般的伺服器，擁有的資源比較多，可將所有 API 都安裝進去，包含 Server 的 API 與 Client 的 API，這樣可以方便開發 Web Service 程式時作測試。

完成之後，原始程式有個 samples 資料夾，裡面會有一些 Web Service 程式範例，可以參考原本的範例程式，呼叫需要的 API 製作自己的 Web Service 程式。

Client 部份因使用嵌入式系統，因此我們須先建立 ARM-Linux 的開發環境，這邊以 FreeBSD 系統為例，大致步驟如下：

1. 安裝 Linux 模擬套件：安裝 /usr/ports/emulators/linux_base-rh-9 套件
2. 安裝 Cross-Compiler：安裝光碟中 PCM-7220-Linux\arm_gcc_and_gdb 資料夾中的 arm_linux_gcc.tar.gz，可參考該資料夾的 readme.txt 進行，解壓縮後 2.95.3/arm-linux/lib 為嵌入式系統的 lib 檔案，2.95.3/arm-linux/include 為嵌入式系統的 include 檔案，此目錄類似 FreeBSD 系統或是 Linux 系統

的/usr/include，在之後的開發過程中，若有缺少系統的 include 而發生編譯錯誤的話，可從這邊取得。

3. 接著修改.bash_profile 檔案，增加路徑 PATH=/path/to/arm-linux-gcc :
\$PATH，接著重新登入主機。
4. 接著測試是否可以從命令列找到 arm-linux-gcc 這個程式，無法找到的話，檢查第 2.步驟與第 3.步驟是否正確。
5. 安裝 OpenSSL 套件：由於套件需要改為嵌入式系統的格式，因此需到 openssl 網站 <http://www.openssl.org/source/> 下載最新原始檔案，依序執行
./Configure --prefix=/build
--openssldir=/home/ailly/arm/armtools/arm-linux/include linux-elf-arm 與
make CC=arm-linux-gcc F77=arm-linux-g77 LD=arm-linux-ld
AR="arm-linux-ar r" RANLIB=arm-linux-ranlib STRIP=arm-linux-strip，最後 make install 若有編譯的錯誤可參考該套件說明檔案，把 Makefile 修改成對應嵌入式系統的參數。
6. 安裝 libxml 以啟動系統處理 XML 檔案的 lib，一樣需要下載原始程式修改，可從 <http://xmlsoft.org/sources/> 下載到，執行 CC=arm-linux-gcc
AR=arm-linux-ar RANLIB=arm-linux-ranlib
STRIP=arm-linux-strip ./configure --host=arm-linux --prefix=/build
--enable-add-ons --with-headers=/home/ailly/armtools/arm-linux/include，產生 Makefile，接著執行 make，成功後再 make install，若遇到問題可以嘗試修改 config.status 檔案，修改後重新執行 config.status 再編譯。

7. 安裝 OpenSOAP 程式，到

<http://www.opensoap.jp/download/opensoap-2.0.1.tar.gz> 下載原始程式，解

壓縮之後，執行 `CC=arm-linux-gcc AR="arm-linux-ar" F77=arm-linux-g77`

`LD=arm-linux-ld RANLIB=arm-linux-ranlib`

`STRIP=arm-linux-strip ./configure --target=arm-linux`

`--build=i386-unknown-freebsd4.9`

`--with-ssl=/home/ailly/arm/armtools/arm-linux --disable-system-iconv`

`--with-apxs=no --with-headers=/home/ailly/arm/armtools/arm-linux/include`

`--prefix=/build/opensoap`，接著修改 `config.status`，修改後重新執行

`config.status` 再編譯。或將本研究修改過的 `config.status` 複製到原始檔資料

夾，做適當修改後，執行 `config.status` 產生 `Makefile`，接著編譯，因 Client

端要放置於嵌入式系統中，因此只需要安裝需要的部份，首先到 `src/api`

執行 `make libOpenSOAPClient.la`，接著 `make install`，會編譯出

`libOpenSOAPClient.la` 與 `.libs/libOpenSOAPClient.so.2`，Client 需要用到的

環境只有 `libOpenSOAPClient.so.2`。

8. 在 `libxml2` 安裝的目錄中會有個 `lib` 目錄，例如安裝到 `/build` 的話，在

`/build/lib` 中有個 `libxml2.so.2.6.19`，將這個檔案複製到嵌入式系統中相同

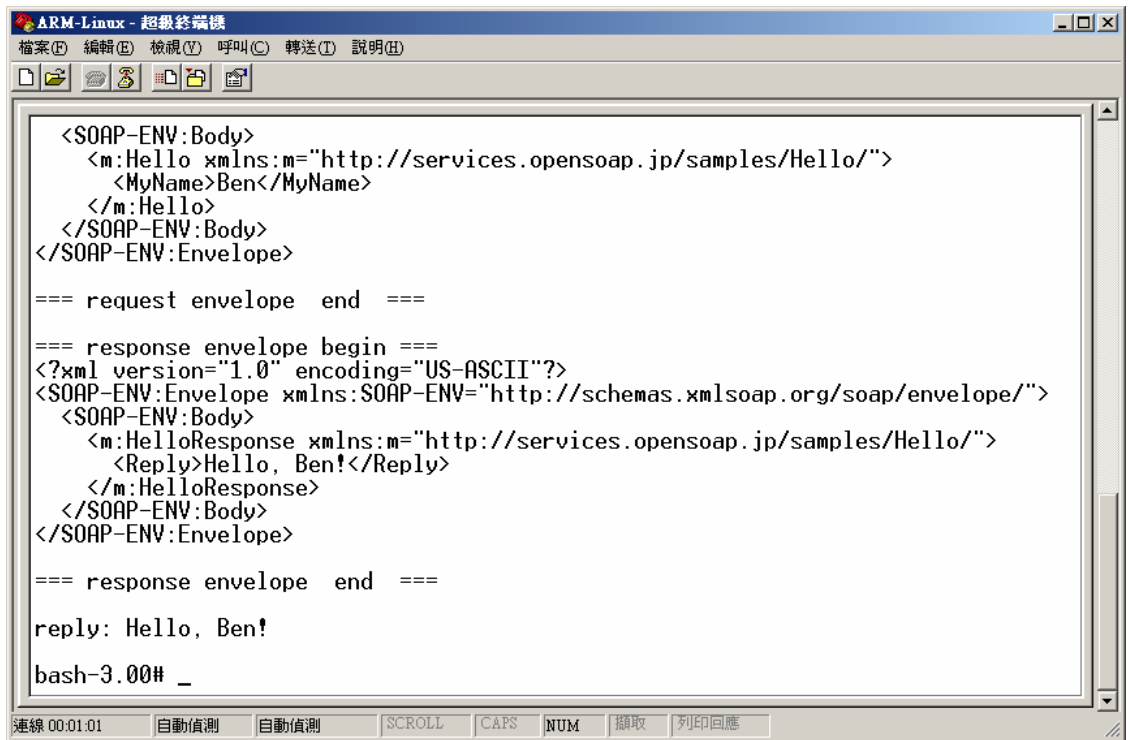
的位置，並且將檔案名稱改為 `libxml2.so.2`，接著將 `/build/opensoap/lib/`

`libOpenSOAPSvc.so.2` 複製到嵌入式系統中一樣的路徑。

完成後即可於開發環境中修改 Web Service 程式，編譯時只需要編譯 Client 的部份，例如 `samples/Hello` 就只需要 `make HelloClient`，編譯完成後 `.libs` 目錄中會出現檔案名稱為 `HelloClient` 的程式，將此程式複製到嵌入式系統中即可執行，

如果沒有問題，該 HelloClient 程式在嵌入式環境中執行的結果如下【圖三十八】

所示：



```
ARM-Linux - 超級終端機
檔案(F) 編輯(E) 檢視(V) 呼叫(C) 轉送(T) 說明(H)

<SOAP-ENV:Body>
  <m:Hello xmlns:m="http://services.opensoap.jp/samples/Hello/">
    <MyName>Ben</MyName>
  </m:Hello>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

=== request envelope end ===

=== response envelope begin ===
<?xml version="1.0" encoding="US-ASCII"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <m:HelloResponse xmlns:m="http://services.opensoap.jp/samples/Hello/">
      <Reply>Hello, Ben!</Reply>
    </m:HelloResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

=== response envelope end ===

reply: Hello, Ben!

bash-3.00# _
```

【圖三十八】 範例程式在嵌入式系統執行結果

Middleware 與 Web Service 的開發平台完成後，再來就可以很簡單的新增高速公路即時路況的通報功能，不需要太多複雜的設定。使用者的部份，呼叫 API 功能實作一個 Web Service Client 程式，使用 SOAP 訊息傳遞目前所在的高速公路位置。伺服器端程式依據前面 Information Fetcher 取得的 XML 檔案資料做處理得知該使用者位置即時的北上與南下時速。接著呼叫 API 將結果轉為 SOAP 訊息回覆該使用者即可完成。於嵌入式系統執行路況查詢 80 公里位置，所得之結果如下【圖三十九】所示：

```
ms.im.ncnu.edu.tw - PuTTY
日期：94/06/23時間：15:31:00地點：國道1號 方向：北 開始地點：326 --*排除*--過?
紡w交流道內移動性施工
日期：94/06/23時間：14:42:00地點：國道1號 方向：北 開始地點：345.5 **1606施工單位來電撤工了**15.15定點**過岡山收費站封閉外側修補路面-施工單位-(終於要修了...)</Reply>
</m:FreewayResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

=== response envelope end ===
執行結果：
North is: 85
South is: 85
info:
日期：94/06/23時間：16:16:00地點：國道1號 方向：南 開始地點：348.5 岡山交流道前2車事故
日期：94/06/23時間：16:08:00地點：國道1號 方向：南 開始地點：348.5 岡山交流道前定點
日期：94/06/23時間：15:31:00地點：國道1號 方向：北 開始地點：326 --*排除*--過?
紡w交流道內移動性施工
日期：94/06/23時間：14:42:00地點：國道1號 方向：北 開始地點：345.5 **1606施工單位來電撤工了**15.15定點**過岡山收費站封閉外側修補路面-施工單位-(終於要修了...)</Reply>
ftp [/home/ailly/opensoap-2.0.1/samples/Freeway] -root- █
```

【圖三十九】 在嵌入式系統執行路況查詢之結果